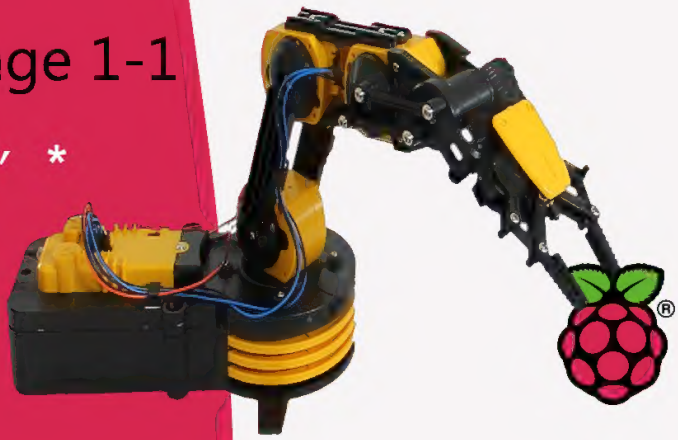
FUZE

GET WITH THE PROGRAM

Project Card Special Stage 1-1

"Wir sind die Roboter" *



Written by Jon Silvera

©2014 **FUZE BASIC**, **FUZE BASIC** Project Cards and the **FUZE** logo are copyright FUZE Technologies Ltd. This document may be copied and distributed within educational establishments, however it may not be edited, modified or sold in any format or by any method without written approval from FUZE Technologies Ltd. For more information about **FUZE BASIC** Project Cards and **The FUZE** visit www.fuze.co.uk. Raspberry Pi and the Raspberry Logo are trademarks of the Raspberry Pi Foundation and are used with permission. All rights reserved.

Raspberry Pi Edition

Special Project 1-1 - The Robots

Before you attempt this project it is important to have covered through some of the early steps of using **FUZE BASIC**. Please be sure to have completed Projects 1-1 and 1-2 before starting this one. Please setup your FUZE computer and connect the Robot Arm to one of the available USB ports. It's best to connect the Arm before running FUZE BASIC and make sure it's switched on.

What do you mean you haven't built the Robot Arm yet? Well that just won't do will it. Ok off you go then and come back once you've done it, in about three hours I reckon.

Double click the FUZE BASIC icon to begin

As you will have come to expect, FUZE BASIC will leap into action and present you with the Ready> prompt.

First of all straighten the robot arm a bit so it's not all folded up. See **[Pic 2]**. Don't worry if the arm clicks here and there this is just the gears clicking and nothing actually breaking.

Type in;

`ArmBody (1)` - press Enter
`ArmBody (-1)` - press Enter
`ArmBody (0)` - press Enter

Warning: These commands will set the robot arm moving as soon as you press enter. If you don't type the next command the arm will go as far as it can and start clicking - you should enter the `ArmBody (0)` command to stop it.

If at this point you get an error stating "unable to find Robot Arm" or similar then exit FUZE BASIC using `Exit` - press Enter

Ha, Robots eh… they'll never take off. Mind you they said that about flying machines once.

GET WITH THE PROGRAM

Project Card Special Stage 1-1

"Wir sind die Roboter"

Robota

The first known use of the word Robot comes from the Czech "Čapek" brothers and was used in Karel Čapek's science fiction play "Rossum's Universal Robots"

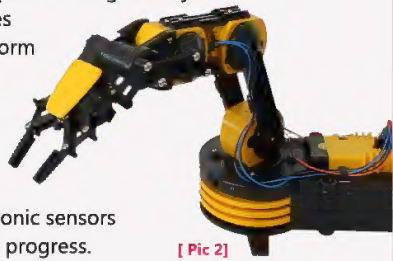
The original meaning of the word Robota, was "drudgery" or "slave labour".

How awful is that!

[Pic 1]

Robots and computing

Whereas early robotic equipment was generally based on complex machines cleverly engineered to perform a specific labour saving function, it was not until the development of integrated circuits (silicon chips) and small powerful motors and electronic sensors that things really started to progress.



[Pic 2]

My favourite robot by far is "Marvin the Paranoid Android" by Douglas Adams and his comedy science fiction story The Hitchhiker's guide to the galaxy. Marvin is a very, very depressed robot and well known for letting us know this with quotes like; *"My capacity for happiness," he added, "you could fit into a matchbox without taking out the matches first"*

Unplug the Arm and reconnect it again. Also please make sure the Arm is switched on **[Pic 3]**. Start **FUZE BASIC** again try the above again. If at this point it doesn't work, seek help from an adult or if you *are* an adult, from a child. On the basis things did work, try the same with these other control commands;

`ArmShoulder (x)` - x can be 1, -1 or 0
`ArmElbow (x)` - x can be 1, -1 or 0
`ArmWrist (x)` - x can be 1, -1 or 0
`ArmGripper (x)` - x can be 1, -1 or 0
`ArmLight (x)` - x can be 1 or 0

A useful trick to know at this point is that you can repeat the last command by pressing the up arrow key and then just edit the number. **Remember though**, you still need to press enter.

Let's put some of this new found knowledge into action. Press **F2** to enter the Editor. If there's another program listed then make sure it isn't needed and then press **F12** to clear it.

Start with the following lines of code;

Press **F3** to run the program. You will be prompted for a file name. Best to name it something like **"JSrobot"** where **JS** is your initials so you don't overwrite someone else's program.

Editor

```
CLS
PROC ResetArm
END

DEF PROC ResetArm
ArmBody (0)
ArmShoulder (0)
ArmElbow (0)
ArmWrist (0)
ArmGripper (0)
ArmLight (0)
ENDPROC
```

The purpose of this section is to make sure the Arm can be instructed to switch everything off, so absolutely nothing will happen at this point but we will use this bit a lot later.

Leonardo da Vinci designed and built the first known humanoid robot around 1495.

GET WITH THE PROGRAM

Project Card Special Stage 1-1

"Wir sind die Roboter"

[Pic 3]

Today, robots can be incredibly complicated machines. Even a small toy robot can have in excess of twenty motors, light sensors, cameras, speech synthesis and significant computing power, all in a device six inches tall!

Do androids dream of electric sheep?

There are so many questions surrounding robotics but perhaps none more so than those concerned over the future of the human race. Are we developing machines that will one day become intelligent and then decide they should rule the world?

Some people believe that as mankind evolves we will enhance ourselves more and more with robotic parts and electronic circuitry. This is known as Cybernetics. The result, the Cyborg or for a simpler term, The Man Machine!

Where will it end... with the complete annihilation of the human race of course, boo!

Not to worry though as by then we will have transferred our consciousness's into computers and as such will live forever and be permanently connected to the Internet with ultra fast broadband... yay!

GET WITH THE PROGRAM

Project Card Special Stage 1-1

"Wir sind die Roboter"

Edit the program to add the following - the grey text is what you should already have. You can see how it should look in **[Pic 4]**

Editor press F2 to display the editor

```
PROC ResetArm
PROC DisplayInstructions
END

DEF PROC ResetArm
ArmBody (0)
ArmShoulder (0)
ArmElbow (0)
ArmWrist (0)
ArmGripper (0)
ArmLight (0)
ENDPROC

DEF PROC DisplayInstructions
CLS
FONTSCALE (2, 2)
INK = Red
PRINT "We are the ROBOTS!"
INK = White
HVTAB (0,2)
PRINT "Press"
PRINT
PRINT "1 or 2 for Body left & right"
PRINT "3 or 4 for Shoulder up & down"
PRINT "5 or 6 for Elbow up & down"
PRINT "7 or 8 for Wrist up & down"
PRINT "9 or 0 for Gripper open & close"
PRINT "Enter to turn the Robot light on"
INK = Red
PRINT
PRINT "Space to stop movement & switch light off"
ENDPROC
```

The robot was an armoured knight that could sit up, wave its arms, and move its head while opening and closing its jaw, presumably meant to scare children who were misbehaving. Sounds like just about every Head Teacher I've ever met!

[Pic 4]

```
PROC ResetArm
PROC DisplayInstructions
END

DEF PROC ResetArm
ArmBody (0)
ArmShoulder (0)
ArmElbow (0)
ArmWrist (0)
ArmGripper (0)
ArmLight (0)
ENDPROC

DEF PROC DisplayInstructions
CLS
FontScale (2, 2)
INK = Red
PRINT "We are the ROBOTS!"
INK = White
HvTab (0, 2)
PRINT "Press"
PRINT
PRINT "1 or 2 for Body left & right"
PRINT "3 or 4 for Shoulder up & down"
PRINT "5 or 6 for Elbow up & down"
PRINT "7 or 8 for Wrist up & down"
PRINT "9 or 0 for Gripper open & close"
PRINT "Enter to turn the Robot light on"
INK = Red
PRINT
PRINT "Space to stop movement & switch light off"
ENDPROC
```

```
We are the ROBOTS!

Press

1 or 2 for Body left & right
3 or 4 for Shoulder up & down
5 or 6 for Elbow up & down
7 or 8 for Wrist up & down
9 or 0 for Gripper open & close
Enter to turn the Robot light on

Space to stop movement & switch light off

x F2 -> Edit or ESC:
```

At the moment this just displays the text as listed in the program. We still need to add the best bit - brrrr... click ..... Brr click .... Brrrrr... soon, soon, do be so impatient!

We have introduced a couple of new commands deserving a brief explanation.

PROC DisplayInstructions and PROC ResetArm

The **PROC** command is short for Procedure. The command tells the program to jump to the part of the program labelled **DEF PROC** "**procedure name**"; in this case **DisplayInstructions** and **ResetArm**.

The end of the procedure is defined by the **ENDPROC** or End Procedure command at which point the program will return to where it was called from.

Procedures help keep a program really tidy as we can place routines and functions away from the main program. They also allow us to reuse the same routine many times with a single command. The **ResetArm** procedure for example can be used at any point to turn everything off just by using **PROC ResetArm**. It's important to grasp this as we will be using them a lot later.

FONTSCALE is very straightforward (1, 1) is normal size whereas (3, 3) is three times width and height and (2, 4) is double width but four times the height. You can experiment with this in Direct mode.

HVTAB is also very simple to grasp once explained. H is for Horiziontal and V is for Vertical. The command positions the text cursor at a specified position on the screen so that the next **PRINT** command will place the text at that position on screen. See opposite.

2040 is the year we should be in fear of, or looking forward to - you decide…

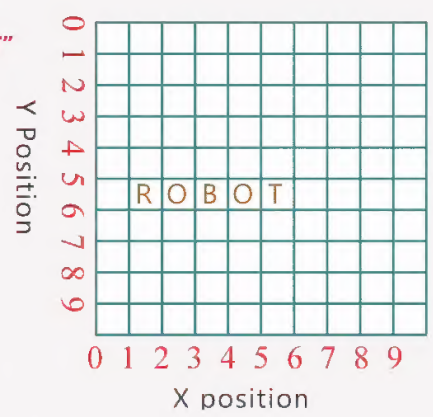
Screen coordinates

There are two ways to display information on the screen using **FUZE BASIC** or just about any other language for that matter. We either place text on the screen or graphics and pictures.

In both cases we us a simple X and Y coordinate system. When using text, the X & Y positions are determined by the width of the text so if you use huge text you will have very large grid spaces.

The example below is made possible by using the following HVTAB command;

```
HVTAB (1, 5)
PRINT "ROBOT"
```



However when we use graphics the X & Y coordinates are based on the height and width of the screen. Also the Y position 0 starts at the bottom left, not at the top as when using text.

Editor

```
PROC ResetArm
PROC DisplayInstructions

CYCLE

Key = Inkey

SWITCH (Key)
   CASE 49
    ArmBody (1)
   ENDCASE
   CASE 50
    ArmBody (-1)
   ENDCASE
   CASE 51
    ArmShoulder (1)
   ENDCASE
   CASE 52
    ArmShoulder (-1)
   ENDCASE
   CASE 53
    ArmElbow (1)
   ENDCASE
   CASE 54
    ArmElbow (-1)
   ENDCASE
   CASE 55
    ArmWrist (1)
   ENDCASE
   CASE 56
    ArmWrist (-1)
   ENDCASE
```

There is a lot to add here so please be sure to copy it exactly. Notice you need to continue typing the second column right after the first.

Editor

```
   CASE 57
    ArmGripper (1)
   ENDCASE
   CASE 48
    ArmGripper (-1)
   ENDCASE
   CASE 32
    PROC ResetArm
   ENDCASE
   CASE 13
    ArmLight (1)
   ENDCASE
ENDSWITCH

REPEAT

END

DEF PROC ResetArm
ArmBody (0)
ArmShoulder (0)
ArmElbow (0)
ArmWrist (0)
ArmGripper (0)
ArmLight (0)
ENDPROC
```

Some scientists and computer futurists predict that robots will emerge as their own species by 2040, with feelings and expectations.

More 'stuff' about robots...

The Three Laws of Robotics

The extremely renowned and very prolific science fiction author Isaac Asimov wrote hundreds upon hundreds of books, articles and short stories about science and robotics.

He introduced the idea of programming a set of rules or laws into all robots to protect humankind.

The stories written around these laws are extremely popular.

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.

2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Asimov later went on to add a new law to precede these - the "Zeroth Law", which focuses on humanity as a whole rather than the individual.

0. A robot may not harm humanity, or, by inaction, allow humanity to come to harm.

There are many debates as to whether we should impliment a similar set of rules into modern day robots.

Personally, I think we'll need something, otherwise who knows what might happen.

Editor

```
DEF PROC DisplayInstructions
CLS
FONTSCALE (2, 2)
INK = Red
PRINT "We are the ROBOTS!"
INK = White
HVTAB (0,2)
PRINT "Press"
PRINT
PRINT "1 or 2 for Body left & right"
PRINT "3 or 4 for Shoulder up & down"
PRINT "5 or 6 for Elbow up & down"
PRINT "7 or 8 for Wrist up & down"
PRINT "9 or 0 for Gripper open & close"
PRINT "Enter to turn the Robot light on"
INK = Red
PRINT
PRINT "Space to stop movement & switch light off"
ENDPROC
```

So two new things here; firstly the **Inkey** statement. This is a very useful command indeed. Please read the opposite page - "More about **Inkey**".

In our program we are storing the value of **Inkey** (the code value of any key pressed) in the variable **Key**

The rest is much easier than it looks. The **SWITCH** and **CASE** commands check the value stored in **Key** and depending on the value performs the command(s) in the relevant **CASE** section.

So if the "**1**" is pressed, the code value is **49** (see reference chart opposite) and so the command, **ArmBody (1)** is excecuted.

It's always nice to have something to look forward to!

More about Inkey

The Inkey command is a very important one that you will use over and over again.

For example, we can use Inkey to pause any program to wait (LOOP) for a key to be pressed;

```
PRINT "Press any key to continue"
CYCLE
REPEAT UNTIL Inkey <> -1
```

If no key is being pressed the the value of **Inkey** is -1. Whenever a key is pressed its code value is stored in **Inkey**. So the above loop will repeat until Inkey is not equal to -1.

This also means we can check if a specific key is pressed. For example the value of the space bar is 32 so we could change the above to;

```
PRINT "Press the Space bar to continue"
CYCLE
REPEAT UNTIL Inkey = 32
```

This time the program waits specifically for the space bar to be pressed and anything else will be ignored.

Here's a few more Inkey codes, just in case you need them.

48 - 0	49 - 1	50 - 2	51 - 3	52 - 4	53 - 5
54 - 6	55 - 7	56 - 8	57 - 9	65 - a	66 - b
67 - c	68 - d	69 - e	70 - f	71 - g	72 - h
73 - I	74 - j	75 - k	76 - l	77 - m	78 - n
79 - o	80 - p	81 - q	82 - r	83 - s	84 - t
85 - u	86 - v	87 - w	88 - x	89 - y	90 - z
32 - Space Bar		13 - Enter			

Special Stage 1-1 Assessments

Time to prove you've mastered the Robot Arm Project

Assessment 1 ⭐

Practice moving different parts of the robot around in Direct mode? Remember **F2** switches between Direct mode and the Editor.

Assessment 2 ⭐

Can you write a different program to display (PRINT) differerent things at different locations on the screen. Remember to Save your work before entering the **NEW** command to start a new program.

Assessment 3 ⭐⭐⭐

Write a new program to repeat a series of Robotic movements. Use the **WAIT** command to determine how far each movement goes.

End of Special Stage 1-1

*The album, The Man Machine by the German band Kraftwerk, has a very good song about Robots on it called "We are the Robots", but of course it was originally in German, hence the project title.

GET WITH THE PROGRAM

Project Card Special Stage 1-1

"Wir sind die Roboter"

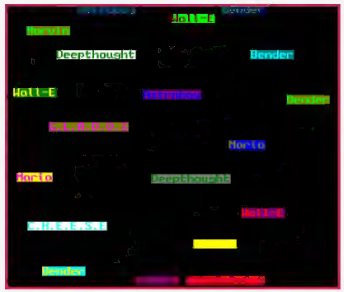
Control commands

The commands are;
ArmBody (x)
ArmShoulder (x)
ArmElbow (x)
ArmWrist (x)
ArmGripper (x)
ArmLight (x)

Where x is 1, -1 or 0
The light is just 1 or 0

HVTAB (x,y)

Remember, **HVTAB** places the cursor at a set position. X is the horizontal and Y is the Vertical.



One more from Marvin;

Marvin: *"I am at a rough estimate thirty billion times more intelligent than you. Let me give you an example. Think of a number, any number."*

Zem: *"Er, five."*

Marvin: *"Wrong. You see?"*

Marvin the paranoid android, by Douglas Adams from The Hitchhiker's Guide to the Galaxy

If you've completed this project, you deserve a pat on the back, perhaps you could program the Robot Arm to do this for you!